



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/786,748	02/25/2004	Rob Woollen	BEAS-01433US1	2991
23910	7590	09/20/2007		
FLIESLER MEYER LLP 650 CALIFORNIA STREET 14TH FLOOR SAN FRANCISCO, CA 94108			EXAMINER YIGDALL, MICHAEL J	
			ART UNIT 2192	PAPER NUMBER
			MAIL DATE 09/20/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/786,748

Applicant(s)

WOOLLEN ET AL.

Examiner

Michael J. Yigdoll

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 01 August 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,4,6-8,10,11,14,16-18,20,21,24,26-28 and 30-42 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1,4,6-8,10,11,14,16-18,20,21,24,26-28 and 30-42 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date 8/1/07.
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on August 1, 2007 has been entered. Claims 1, 4, 6-8, 10, 11, 14, 16-18, 20, 21, 24, 26-28 and 30-42 are now pending.

Response to Amendment

2. The objections to claims 1, 11 and 21 set forth in the Office action mailed on February 1, 2007 have been withdrawn in view of Applicant's amendment.

Response to Arguments

3. Applicant's arguments have been considered but are moot in view of the new ground(s) of rejection, as set forth below.

Specification

4. The use of the trademarks JAVA, JAVA 2 ENTERPRISE EDITION (J2EE), ENTERPRISE JAVABEAN (EJB), JAVA SERVER PAGE (JSP), WEBLOGIC and WEBLOGIC SERVER has been noted in this application (see, for example, specification, paragraphs [0003] and [0010], new paragraph [0007.1], and new claims 34, 37 and 40).

All trademarks should be capitalized (i.e., each letter should be capitalized) wherever they appear and should be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner that might adversely affect their validity as trademarks.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 31-33 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter that Applicant regards as the invention.

With respect to claim 31 (currently amended), the claim recites "the output directory" in line 4, for which there is insufficient antecedent basis in the claims. The examiner's presumption is that the claim should recite --the output folder-- such as recited in claim 1.

With respect to claim 32 (currently amended), the claim recites "the output directory" in line 4, for which there is insufficient antecedent basis in the claims. The examiner's presumption is that the claim should recite --the output folder-- such as recited in claim 11.

With respect to claim 33 (currently amended), the claim recites "the output directory" in line 4, for which there is insufficient antecedent basis in the claims. The examiner's presumption is that the claim should recite --the output folder-- such as recited in claim 21.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1, 4, 6-8, 10, 11, 14, 16-18, 20, 21, 24, 26-28 and 30-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pub. No. 2004/0088681 to Berg et al. (art of record, "Berg") in view of U.S. Pub. No. 2002/0178439 to Rich et al. (art of record, "Rich") and in view of U.S. Patent No. 6,178,546 to McIntyre (art of record, "McIntyre").

With respect to claim 1 (currently amended), Berg discloses a system for organization of software application files during development and subsequent deployment of the software application to a server (see, for example, the abstract), comprising:

a split directory structure stored on a computer medium that stores files for a software application (see, for example, paragraph [0025], lines 1-7, which shows a split directory structure for files of a software application), wherein, for each software application, the split directory structure includes both a source folder that stores editable source files as part of the software application (see, for example, paragraph [0008], lines 1-7, which shows that the split directory structure includes such a source folder), and a corresponding output folder that stores compiled files as part of the software application (see, for example, paragraph [0006], lines 6-11, which shows that the split directory structure includes such an output folder).

Here, the container project of Berg (see, for example, paragraph [0025], lines 1-7) is considered to represent a split directory structure for an application. The container project (i.e., the split directory structure) includes a source folder that stores editable source files (see, for example, paragraph [0008], lines 1-7). The utility JAR of Berg stores compiled files (see, for example, paragraph [0006], lines 6-11), and is considered to represent an output folder at least because its contents are expanded into a directory (see, for example, paragraph [0005], lines 11-16). The utility JAR (i.e., the output folder) is included in the container project (i.e., the split directory structure) and is considered to represent a “corresponding” output folder at least because it corresponds to the same container project as the source folder (see, for example, paragraph [0010], lines 3-14).

Berg discloses virtual archives (see, for example, paragraph [0036], lines 1-7), and further discloses JAR files and other related archives (see, for example, paragraph [0005], lines 5-11 and paragraph [0007], lines 1-8), but does not expressly disclose that the split directory is accessed as a virtual file that provides an abstraction over the two folders therein.

However, in an analogous art, Rich discloses a virtual archive that provides an abstraction over a directory structure (see, for example, paragraph [0019], lines 1-11), and further discloses such an archive in the format of a JAR file (see, for example, paragraph [0005], lines 1-22 and paragraph [0007], lines 1-11). The virtual archive (i.e., the virtual file) provides a simple, common and transparent interface for directory structures and archive files alike (see, for example, paragraph [0018], lines 1-13).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to access the split directory of Berg as a virtual file that provides an

abstraction over the two folders therein, as Rich suggests, so as to access the split directory with a simple, common and transparent interface.

Berg in view of Rich further discloses:

a server upon which the software application will be deployed (see, for example, Berg, paragraph [0031], lines 1-6, which shows an application server); and

a deployment tool that allows a user to specify the output folder during deployment of the software application (see, for example, paragraph [0027], lines 1-3, which shows a deployment tool for specifying an output folder to deploy), wherein during the deployment the server interprets the software application as a union of both the source folder and the output folder, recognizes the split directory structure, and deploys the application by making requests to the virtual file which checks the source folder and the corresponding output folder for software application files (see, for example, Berg, paragraph [0027], lines 3-9 and paragraph [0029], lines 1-5, which shows that the server checks the container project, i.e., the split directory structure that represents a union of the source and output folders, for the software application files, and see, for example, Rich, paragraph [0052], lines 1-17, which shows the manner in which such requests are made to the virtual file), before deploying the software application files to the server (see, for example, Berg, paragraph [0030], lines 1-4).

Berg in view of Rich further discloses checking for and finding the files in any location (see, for example, Berg, paragraph [0039], lines 1-5), but does not expressly disclose that the virtual file checks first the source folder and then the corresponding output folder for software application files.

However, in an analogous art, McIntyre discloses first checking one build area for software application files, and, if the files are not found in the build area, then checking another build area (see, for example, column 2, lines 57-60). This enables a developer to separately update his or her own files while still referencing files from other developers (see, for example, column 2, lines 54-57 and 60-62).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the virtual file of Berg in view of Rich such that it first checks the source folder and then the corresponding output folder for software application files, as McIntyre suggests. Furthermore, it would have been obvious to check the source folder first and then check the output folder because, as Berg discloses, it is optimal for the software application to access the source folder so that the programmer can update those files directly (see, for example, paragraph [0009], lines 6-9), rather than having to update copies of the files stored in the output folder (see, for example, paragraph [0011], lines 1-11).

With respect to claim 4 (previously presented), the rejection of claim 1 is incorporated, and Berg in view of Rich and McIntyre further discloses that the output folder includes a file that identifies the output folder as being part of the split directory which also includes the corresponding source folder (see, for example, Berg, paragraph [0026], lines 1-3, which shows such a file).

With respect to claim 6 (original), the rejection of claim 1 is incorporated, and Berg in view of Rich and McIntyre further discloses that said software application, or another software application can point to the output folder to access or retrieve resources in either the output

Art Unit: 2192

folder and/or the source folder as necessary for operation of the software application (see, for example, Berg, paragraph [0037], lines 1-7, which shows that software applications access resources in the output and/or source folders as necessary).

With respect to claim 7 (original), the rejection of claim 1 is incorporated, and Berg in view of Rich and McIntyre further discloses that said output folder is automatically created and populated upon compiling the software application (see, for example, Berg, paragraph [0006], lines 6-11, which shows that the output folder is created and populated with compiled files, and see, for example, McIntyre, column 4, lines 37-43, which shows an output folder within a build area that stores compiled files for a software application, and column 6, lines 57-60, which shows that the output folder is automatically populated upon compiling the software application).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to automatically create and populate the output folder of Berg in view of Rich and McIntyre upon compiling the software application, as McIntyre suggests, so as to provide the system with further automation.

With respect to claim 8 (original), the rejection of claim 1 is incorporated, and Berg in view of Rich and McIntyre further discloses that said output folder can be deleted to remove the latest build of the software application, and then recreated to create a new build (inherently, the output folder can be deleted and recreated to remove the software application and create a new build).

With respect to claim 10 (original), the rejection of claim 1 is incorporated, and Berg in view of Rich and McIntyre further discloses that the source folder is populated with source files

Art Unit: 2192

that are stored in or retrieved from a source control system (see, for example, Berg, paragraph [0024], lines 1-5, which shows that source folder is populated with files from an IDE or source control system).

With respect to claims 11 (currently amended), 14 (previously presented), 16-18 and 20 (original), the claims are directed to a method that is analogous to the system of claims 1, 4, 6-8 and 10, respectively (see the rejection of claims 1, 4, 6-8 and 10 above).

With respect to claims 21 (currently amended), 24 (previously presented), 26-28 and 30 (original), the claims are directed to a computer readable medium that is analogous to the system of claims 1, 4, 6-8 and 10 (see the rejection of claims 1, 4, 6-8 and 10 above).

With respect to claim 31 (currently amended), the rejection of claim 1 is incorporated, and Berg in view of Rich and McIntyre further discloses that the virtual file first checks the source folder for the software application files including any classes or resources needed by the software application, and, if the classes or resources are not found in the source folder, then checks the output directory (see, for example, Berg, paragraph [0006], lines 6-11 and paragraph [0007], lines 1-8, which shows that the software application files include classes and resources, and see, for example, McIntyre, column 2, lines 57-60, which shows first checking one build area for software application files, and, if the files are not found in the build area, then checking another build area).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the virtual file of Berg in view of Rich and McIntyre such that it first checks the source folder for the software application files including any classes or resources

needed by the software application, and, if the classes or resources are not found in the source folder, then checks the output directory, as McIntyre suggests. Furthermore, it would have been obvious to check the source folder first and then check the output folder because, as Berg discloses, it is optimal for the software application to access the source folder so that the programmer can update those files directly (see, for example, paragraph [0009], lines 6-9), rather than having to update copies of the files stored in the output folder (see, for example, paragraph [0011], lines 1-11).

With respect to claim 32 (currently amended), the claim is directed to a method that is analogous to the system of claim 31 (see the rejection of claim 31 above).

With respect to claim 33 (currently amended), the claim is directed to a computer readable medium that is analogous to the system of claim 31 (see the rejection of claim 31 above).

With respect to claim 34 (new), the rejection of claim 1 is incorporated, and Berg in view of Rich and McIntyre further discloses that the source folder includes Java files from a source control system, and wherein the output folder includes generated Java class files (see, for example, Berg, paragraph [0024], lines 1-5, which shows that source folder includes source files from an IDE or source control system, and paragraph [0006], lines 6-11, which shows that the output folder includes Java class files), and wherein when the server searches for a resource or class for use in deploying software application, it first checks the source folder, and if the class or resource is not found, then checks the output folder (see, for example, McIntyre, column 2,

Art Unit: 2192

lines 57-60, which shows first checking one build area for software application files, and, if the files are not found in the build area, then checking another build area).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the server of Berg in view of Rich and McIntyre such that when the server searches for a resource or class for use in deploying software application, it first checks the source folder, and if the class or resource is not found, then checks the output folder, as McIntyre suggests. Furthermore, it would have been obvious to check the source folder first and then check the output folder because, as Berg discloses, it is optimal for the software application to access the source folder so that the programmer can update those files directly (see, for example, paragraph [0009], lines 6-9), rather than having to update copies of the files stored in the output folder (see, for example, paragraph [0011], lines 1-11).

With respect to claim 35 (new), the rejection of claim 1 is incorporated, and Berg in view of Rich and McIntyre further discloses that the system comprises a different split directory structure and corresponding virtual file for each software application to be deployed to the server (see, for example, Berg, paragraph [0008], lines 1-7, which shows that there are different split directory structures and virtual files for different software applications).

With respect to claim 36 (new), the rejection of claim 1 is incorporated, and Berg in view of Rich and McIntyre further discloses that the source files can be modified by the user and then stored in the source folder, and wherein the server automatically sees the new source files and uses them as necessary in deploying the software application (see, for example, Berg, paragraph [0009], lines 6-9, which shows that the source files can be modified and stored in the source

Art Unit: 2192

folder, and see, for example, paragraph [0027], lines 3-9 and paragraph [0029], lines 1-5, which shows that the server uses the source files as necessary in deploying the software application).

With respect to claims 37-39 (new), the claims are directed to a method that is analogous to the system of claims 34-36, respectively (see the rejection of claims 34-36 above).

With respect to claim 40-42 (new), the claims are directed to a computer readable medium that is analogous to the system of claim 34-36, respectively (see the rejection of claims 34-36 above).

Conclusion

9. The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure (see the attached Notice of References Cited).

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2192

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Michael J. Yigdall
Examiner
Art Unit 2192



mjy